

SIMULATING STAR CLUSTERS WITH THE AMUSE SOFTWARE FRAMEWORK. I. DEPENDENCE OF CLUSTER LIFETIMES ON MODEL ASSUMPTIONS AND CLUSTER DISSOLUTION MODES

ALFRED J. WHITEHEAD¹, STEPHEN L. W. McMILLAN¹, ENRICO VESPERINI², AND SIMON PORTEGIES ZWART³¹ Drexel University, Philadelphia, PA 19104, USA; alf.whitehead@drexel.edu² Indiana University, Bloomington, IN 47405, USA³ Leiden Observatory, Leiden University, 2300-RA Leiden, The Netherlands

Received 2013 February 8; accepted 2013 October 7; published 2013 November 11

ABSTRACT

We perform a series of simulations of evolving star clusters using the Astrophysical Multipurpose Software Environment (AMUSE), a new community-based multi-physics simulation package, and compare our results to existing work. These simulations model a star cluster beginning with a King model distribution and a selection of power-law initial mass functions and contain a tidal cutoff. They are evolved using collisional stellar dynamics and include mass loss due to stellar evolution. After studying and understanding that the differences between AMUSE results and results from previous studies are understood, we explored the variation in cluster lifetimes due to the random realization noise introduced by transforming a King model to specific initial conditions. This random realization noise can affect the lifetime of a simulated star cluster by up to 30%. Two modes of star cluster dissolution were identified: a mass evolution curve that contains a runaway cluster dissolution with a sudden loss of mass, and a dissolution mode that does not contain this feature. We refer to these dissolution modes as “dynamical” and “relaxation” dominated, respectively. For Salpeter-like initial mass functions, we determined the boundary between these two modes in terms of the dynamical and relaxation timescales.

Key words: celestial mechanics – globular clusters: general – methods: numerical

Online-only material: color figures

1. INTRODUCTION

Star clusters are natural laboratories for many astrophysical processes. In the simplest description, cluster stars may be thought of as being (almost) coeval point masses—an N -body system—and their motion traces their mutual gravitation and the possible influence of an external galactic tidal field. In more complex situations, stars evolve, gas may accrete into the cluster, new stars may form out of that gas, and the gas may be expelled from the cluster quickly by supernovae or more slowly by radiation pressure and stellar winds. A typical cluster is subject to several long-term mass-loss processes, including losses due to stellar evolution and removal of the outermost stars by the galaxy’s tidal field. These processes compete with relaxation processes to define the equilibrium state of the cluster.

Setting aside the complexities of intracluster gas, simple models combining a few basic physical processes—stellar dynamics, stellar evolution, and tidal effects—have proved very useful in the study of star clusters. These simulations combine differing treatments of multiple physical processes and must be carefully calibrated to ensure their reliability. Chernoff & Weinberg (1990; referred to as “CW” hereafter) combined a simple stellar evolution (SSE) prescription with Fokker–Planck simulations of stellar dynamics and a highly idealized tidal field to produce a seminal “baseline” set of cluster simulations, starting from King (1966) initial models. This survey and subsequent studies by Fukushige & Heggie (1995), Aarseth & Heggie (1998), and Takahashi & Portegies Zwart (2000; “TPZ” hereafter), using other formulations of stellar evolution and both N -body and Fokker–Planck treatments of stellar dynamics, have resulted in comparative catalogs of parameter space that now serve as tests of any new code.

Part of the purpose of this paper is to validate parts of the Astrophysical Multipurpose Simulation Environment

(AMUSE)⁴ against known results and then to show new applications of the framework to stellar cluster dynamics. AMUSE is a new software framework designed for simulations of dense stellar systems, inspired by the earlier MUSE project described by Harfst et al. (2008) and Portegies Zwart et al. (2009). A detailed technical account of AMUSE is beyond the scope of this article (see McMillan et al. 2012; Portegies Zwart et al. 2013). A summary is presented in Section 2 to provide the reader with some context on the software used.

We set out to test AMUSE against known results, but found that comparing different simulations at any meaningful level of precision is a non-trivial task. In order to accomplish this goal, we employ an N -body stellar dynamics code, several stellar evolution codes, and a simple escaper removal algorithm as the three basic simulation components, and we compare AMUSE with the results of TPZ.

This line of inquiry led to a description of the dissolution modes of King models within a tidal cutoff. We demonstrate that competition between the relaxation, dynamical, and stellar evolution timescales leads to a split between dissolutions dominated by relaxation processes and those dominated by dynamical processes. By sampling the relevant timescales, the boundary is mapped.

We also generate a comparison of different stellar evolution codes linked to the same dynamics code and run against the same initial conditions, demonstrating that the specifics of the choice of stellar evolution recipes are amplified by the stellar dynamics and impact the results of the simulation.

The structure of this paper is as follows. In Section 2, we describe AMUSE and its specific use for the dissolving star cluster problem. This is followed by Section 3, where the physical model, including details of the CW stellar evolution

⁴ <http://amusecode.org>

approximation, is detailed. Section 4 contains the validation of AMUSE runs (Sections 4.1 and 4.2), a study of the consequences of the variance in initial conditions on simulations (Section 4.3), an exploration of the types of dissolution that can disrupt a King model (Section 4.4), and a direct comparison of stellar evolution codes (Section 4.5). Finally, Section 5 summarizes the results and proposes future work.

This paper is the first in a series of papers describing work with AMUSE. In this series, we will lay the groundwork for future studies by demonstrating that AMUSE can reproduce well-known published results. Future work will explore various types of N-body codes (direct integration, tree, etc.), as well as the inclusion of binaries and multiple stars.

The series begins with a relatively simple model (single stars in a cluster using a tidal cutoff) and is intended to progress to more realistic models in later work. It is important to establish the reliability of the AMUSE framework through comparison with existing work. We can demonstrate the utility of the modular framework along the way by conducting comparisons between codes that were prohibitively difficult without it.

2. COMPUTATIONAL FRAMEWORK

Historically, astrophysical simulation codes have been constructed by a single author or by a small group working closely together. The typical course of the development begins with a simple solver for a specific physical problem (for example, an N -body integrator for a collisionless system) and then gradually extends to cover more varied physics (to continue the example, collisional physics or stellar evolution effects might be added). This approach has been very successful, but is limited when it comes time to compare codes and implementations, or to extend a simulation to include a new piece of physics (to continue the example again, radiative transfer processes may need to be included). In the case of comparison, the types of physics studied are tightly coupled to a specific implementation. It is non-trivial to change from one stellar evolution recipe (to give one example) to another, unless the authors of the code have included both recipes. In the case of extension, the team of authors behind the code may need to grow to bring in experts in the newly required fields of physics.

Despite these difficulties, a number of very successful codes have been developed. Among these are the NBODY series of codes (for a review, see Aarseth 1999), GADGET (Springel et al. 2001), FLASH (Fryxell et al. 2000), and STARLAB (Portegies Zwart et al. 2001). Nevertheless, it is becoming clear that the limits of the traditional approach are being reached. In order for new physics to be added to these packages, the programmer (or team of programmers) must be an expert in the new physics being added, as well as in every physical domain already present in the tightly coupled code. This, combined with the difficulty of modifying any existing physics in these packages, limits the effectiveness of further work.

The AMUSE philosophy is to move away from a general-purpose multi-physics “solver” and toward a suite of standardized special-purpose “evolver” modules. Each evolver knows about only a single physical domain and is responsible for advancing a known system state through time by implementing the physics specific to that domain. In particular, an evolver is not expected to take into account any physics outside its own domain in its calculations.

The AMUSE standard defines four physical domains of interest: gravitational dynamics, stellar evolution, hydrodynamics, and radiative transfer. A standard interface to an evolver

is defined for each of these domains. For example, the stellar dynamics interface specifies how particles are communicated to the evolver (added, removed, and updated) and how to make the evolver step forward a given number of time units. Similarly, the stellar evolution interface specifies how to communicate star properties (mass, age, metallicity, etc.) to the evolver and how to make the evolver advance to a specified time.

All evolvers for a given physical domain are accessible within the AMUSE environment through this standard interface. This means that evolvers within a domain are interchangeable. As shown in Section 4, it is possible for a researcher to switch between several stellar evolution models to test the effect of changing the physical approximations used on the behavior of the entire system. The same is true of the other physical domains. This decoupling of the underlying science codes from the simulation logic is powerful. Users who are not experts in the details of the scientific modules can “mix and match” reliable existing work to produce new types of simulations.

Wherever possible, the AMUSE approach is to reuse existing codes instead of writing new ones. This means that many special-purpose stand-alone solvers can be turned into AMUSE modules. The framework provides a quick and easy method for wrapping an existing code in one of the standard interfaces and making it available within the AMUSE environment. The decoupling of science codes is a benefit to code authors, as they now need to produce only a solver for an individual physics domain in order to run a realistic simulation. The other physical domains, in which they may not be experts, can be “borrowed” from the AMUSE community codes directly. At the discretion of the author, such a module may also become part of the AMUSE package distributed on the web to interested researchers. Alternately, it is possible to create a “private” AMUSE module that exists only on the author’s computers.

In order to make use of AMUSE, the researcher writes a “top-level” script (using the Python scripting language) that instantiates a set of evolvers relevant to the problem being studied. All communication and synchronization between the evolvers are handled by this script. In this work, the top-level script creates a stellar dynamics evolver (in our case, an N -body code) and a stellar evolution evolver. It then begins a loop in which dynamics and evolution are advanced in tandem, with synchronization between them as needed. It also implements a tidal cutoff by removing escapers from the simulation at fixed time intervals.

AMUSE uses the message-passing interface (MPI; see, for example, Walker 1994) to allow each evolver to exist in its own process, possibly in parallel to and on a different machine than the controlling Python script. Each evolver is written in the language of choice of its original author. Already present in AMUSE are modules written in C, C++, Python, Fortran, and Java. MPI was chosen based on the experience of MUSE (which used SWIG and f2py instead) and allows both for parallelization and for each module to reside in its own unique namespace. AMUSE is compatible with OpenMPI or with MPICH2, or variants thereof. In this work, we have used the MVAPICH2⁵ implementation of MPI because it supports the Infiniband networking present on our GPU computing cluster. AMUSE is also capable of running on a grid for massively parallel calculations (Drost et al. 2012).

Table 1 lists the specific AMUSE modules used in this work. The PH4 evolver provides N -body dynamics using

⁵ <http://mvapich.cse.ohio-state.edu/overview/mvapich2/>

Table 1
AMUSE Modules Used in This Work

Module	Type	Reference
PH4	<i>N</i> -body dynamics	McMillan et al. (2012)
SAPPORO	GRAPE emulator on GPU	Gaburov et al. (2009)
SSE	Stellar evolution	Hurley et al. (2000)
EFT89	Stellar evolution	Eggleton et al. (1989, 1990)
VSSE	Stellar evolution	Chernoff & Weinberg (1990)
SEBA	Stellar evolution	Portegies Zwart & Verbunt (1996)

SAPPORO, a GRAPE emulator, for GPU acceleration. PH4 is an MPI-parallel, adaptive block time-step, GRAPE-accelerated, fourth-order Hermite integrator, similar in construction to PH-GRAPe (Harfst et al. 2007). SSE provides stellar evolution. KNNCUDA (Garcia et al. 2008; Garcia 2008) is used to compute densities (using 12th nearest neighbors) in a stand-alone code similar to that described by Casertano & Hut (1985), but separate from AMUSE. This code uses an exact algorithm to find all nearest neighbors, regardless of distance.

All modules used in this work were compiled for a 64-bit architecture and use double precision floating point variables. AMUSE provides unit conversion features to link the *N*-body code (using dimensionless units) to the stellar evolution code (using physical units). This linkage occurs only in the time and mass variables, and a brief analysis shows that no significant numerical error is expected.

While we have used SAPPORO, which in turn uses CUDA, to emulate a GRAPE hardware accelerator, it is important to note that this is not the only possible choice. An actual GRAPE could be used, as could the SAPPORO_LIGHT module included in AMUSE, which can run with or without a GPU present. Future versions of SAPPORO will use OpenCL, which will allow the use of the AMUSE packages on AMD-based GPUs and other devices supporting that open standard.

Additionally, the EFT89 module, the SEBA module, and the Very Simple Stellar Evolution (VSSE) module were used to provide the simplified stellar evolution models explored in Section 4. VSSE was written specifically for this work and is designed to allow a researcher to easily add simple analytical stellar evolution models, here the prescription of CW (see also Section 3.3).

The cluster was allowed to evolve dynamically using the adaptive, block time-step algorithm embedded in PH4. At regular intervals of 1 Myr (chosen to resolve mass-loss processes in the most massive stars; see Portegies Zwart et al. 1999), the stellar evolution mass loss was computed and applied to the synchronized *N*-body model. The number of synchronizations per dynamical time depends on the dynamical timescale of the initial model. The simulation was stopped, and the cluster considered dissolved, if fewer than 12 stars remained in the cluster.

The specific software architecture used in this work is sketched in Figure 1. Note that GPU acceleration is used for both *N*-body dynamics and the nearest-neighbor calculation.

Our runs were conducted on a cluster of 24 dual-socket Intel Xeon X5650, 2.66 GHz nodes with a total of 12 cores each. Each node contains 64 GB of RAM and six nVidia GTX 480 GPUs. Our runs ran on a single node, using two GPUs: one for dynamics calculation and one for density estimation. A typical run would use up to three cores: one each for process control, dynamics supervision, and stellar evolution.

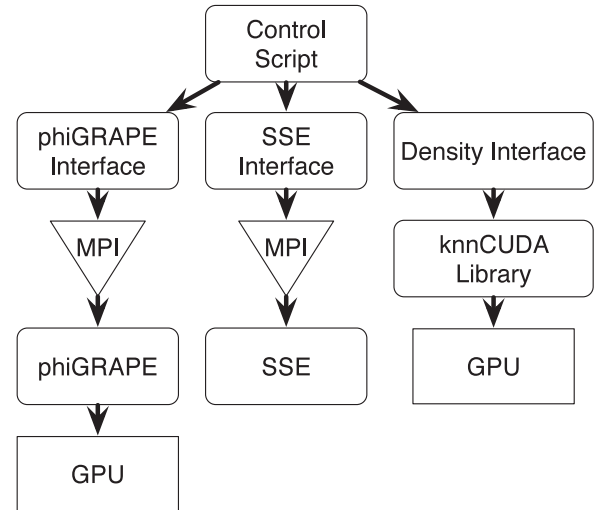


Figure 1. AMUSE software architecture, showing the particular modules employed for the bulk of this work and the hardware acceleration used.

The simulations with PH4 running on SAPPORO using NVidia GPUs perform comparably to GPU-accelerated packages such as GADGET 2 and STARLAB using SAPPORO. A detailed review of the performance of AMUSE simulations can be found in Portegies Zwart et al. (2013). Energy and angular momentum are conserved to within one part in 10^{-7} in test runs using PH4. Energy conservation within AMUSE is studied in more detail in Pelupessy et al. (2013).

3. PHYSICAL MODEL

3.1. Initial Conditions

King (1966) models with $W_0 = 3$ and $W_0 = 7$ were used as initial conditions for our model clusters, representing relatively diffuse and relatively centrally concentrated systems. We obtain the King models by numerical integration, as shown in Section III of King (1966).

A simple power-law stellar mass function

$$dN \propto m^{-\alpha} dm \quad (1)$$

was used in conjunction with a random number generator to assign masses to each particle. Following CW, the slope of the mass function was taken to be either $\alpha = 1.5$ or $\alpha = 2.5$ with masses ranging from $0.4 M_\odot$ to $15 M_\odot$. A Salpeter mass function would correspond to $\alpha = 2.35$ (Salpeter 1955). For $\alpha = 1.5$, we expect approximately 15% of stars to undergo a core collapse supernova. For $\alpha = 2.5$, the supernova fraction is approximately 2%. The assumption of formation by violent relaxation was made. That is to say that the initial mass, position, and velocity of a star are uncorrelated, beyond the condition that the cluster begins in virial equilibrium.

Runs are grouped by “family,” a parameter also defined by CW. The family parameter fixes the relaxation time at the tidal radius, which effectively changes the ratio of the stellar evolution and dynamical timescales for a given model. Four values of this relaxation time are chosen, as summarized in Table 2. The relaxation time at the tidal radius is

$$t_{\text{rlx,CW}} = \frac{M^{1/2} r_t^{3/2}}{G^{1/2} m_\star \ln N} = (2.57 \text{ Myr}) F_{\text{CW}}, \quad (2)$$

where r_t is the King truncation radius and m_\star is a typical stellar mass, here taken to be M_\odot (see Equation (6) of CW,

Table 2
Family Summary

Family	R_g^a (kpc)	F_{CW}	$t_{rlx,CW}$ (Gyr)	t_{rh} (Gyr)			
				$\alpha = 1.5$		$\alpha = 2.5$	
				$W_0 = 3$	$W_0 = 7$	$W_0 = 3$	$W_0 = 7$
1	16.0	5.00×10^4	128	1.01	0.300	2.46	0.728
2	42.1	1.32×10^5	339	2.68	0.793	6.49	1.92
3	71.8	2.25×10^5	577	4.56	1.35	11.1	3.28
4	189	5.93×10^5	1522	12.0	3.56	29.2	8.64

Notes. ^a The value of R_g is slightly different for each random realization, as it is chosen to fix the value of $t_{rlx,CW}$. A typical value is quoted here.

Table 3
Relevant King Model Parameters

W_0	c	r_h/r_t	$\rho(0)/\langle\rho\rangle$
3	0.67	0.27	84
7	1.53	0.12	6430

or Equation (8) of TPZ). By making the assumption that r_t is equal to the Jacobi radius r_J of the cluster, F_{CW} is defined to be

$$F_{CW} \equiv \frac{M}{M_\odot} \frac{R_g}{\text{kpc}} \frac{220 \text{ km s}^{-1}}{v_g} \frac{1}{\ln N}, \quad (3)$$

where R_g and v_g are the radius and velocity of the cluster, assuming a circular orbit about a parent galaxy (see CW). In generating our initial conditions, the choice $v_g = 220 \text{ km s}^{-1}$ is made and R_g is computed to match one of the specified families.

The more conventional half-mass radius relaxation time is given by

$$t_{rh} = 0.138 \frac{M^{1/2} r_h^{3/2}}{G^{1/2} \langle m \rangle \ln N} = (3.54 \times 10^5 \text{ yr}) F_{CW} \left(\frac{r_h}{r_t} \right)^{3/2} \frac{M_\odot}{\langle m \rangle}, \quad (4)$$

where $\langle m \rangle$ is the average mass of a star in the cluster (TPZ). For $\alpha = 1.5$ or 2.5 , $\langle m \rangle = 2.54 M_\odot$ or $1.01 M_\odot$, respectively. The properties of each family are listed in Table 2. For convenience, a brief summary of the relevant King model parameters is also provided in Table 3.

The initial conditions were generated using STARLAB tools and are stored in the STARLAB format, which is AMUSE compatible.

In this first set of simulations performed with AMUSE, we have not considered primordial binary stars. Furthermore, we perform no special treatment of binary or multiple stars in the code, as the version of AMUSE used in most of this work did not support such handling. A softening length of $\epsilon = 1/N$ (roughly the 90 degree turnaround distance for an equilibrium system) was adopted to avoid numerical issues due to close encounters and to avoid hard binary formation. Taking into account the fact that our models cover only the pre-core collapse regime, during which binaries are not expected to form, this omission has no impact on our results. Binary and multiple dynamics are extremely important during core collapse and afterward and will be the subject of Paper II in this series, using capabilities subsequently added to AMUSE.

Table 4
VSSE Lifetimes

Initial Mass ($\log_{10}[m_i/M_\odot]$)	Main-sequence Lifetime ($\log_{10}[t_{se}/\text{yr}]$)
1.79	6.50
1.55	6.57
1.33	6.76
1.11	7.02
0.91	7.33
0.72	7.68
0.54	8.11
0.40	8.50
0.27	8.90
0.16	9.28
0.07	9.63
-0.01	9.93
-0.08	10.18

3.2. Tidal Truncation

Tides are simulated by simple truncation at the Jacobi radius

$$r_J = \left(\frac{M}{3M_{\text{galaxy}}} \right)^{1/3} R_g, \quad (5)$$

which is assumed to be equal to the King truncation radius r_t . Once every dynamical time, any stars that pass beyond the Jacobi radius are removed from the simulation. The dynamical time for the cluster is

$$t_{\text{dyn}} = \frac{GM^{5/2}}{(-2U)^{3/2}}, \quad (6)$$

where U is the gravitational potential energy of the system (Portegies Zwart et al. 2010).

3.3. Very Simple Stellar Evolution (VSSE)

For direct comparison with CW and TPZ, we implemented the Chernoff & Weinberg stellar evolution model in AMUSE. In this model, a main-sequence star remains at constant mass until it has exceeded its lifetime. At that point, the star is transformed into a remnant of lower mass, and evolution ceases.

The stellar lifetime is determined by fitting a cubic spline to the data points listed in Table 4, which were taken by Chernoff & Weinberg from Miller & Scalo (1979). The remnant mass is derived using Equation (7), which was based on Iben & Renzini (1983). Note that the lifetimes listed are assumed to correspond to Population I stars:

$$m_f = \begin{cases} 0.58 M_\odot + 0.22 (m_i - M_\odot), & m_i < 4.7 M_\odot \\ 0.0, & 4.7 M_\odot \leq m_i \leq 8.0 M_\odot \\ 1.4 M_\odot, & 8.0 M_\odot < m_i \leq 15.0 M_\odot \end{cases}. \quad (7)$$

4. RESULTS AND DISCUSSION

4.1. Validation

We first set out to test whether or not the AMUSE framework could reproduce known results. CW produced a well-studied catalog of Fokker–Planck code simulations. TPZ performed runs for the same initial parameters using both a Fokker–Planck code and an N -body code, calibrating the removal of escapers in

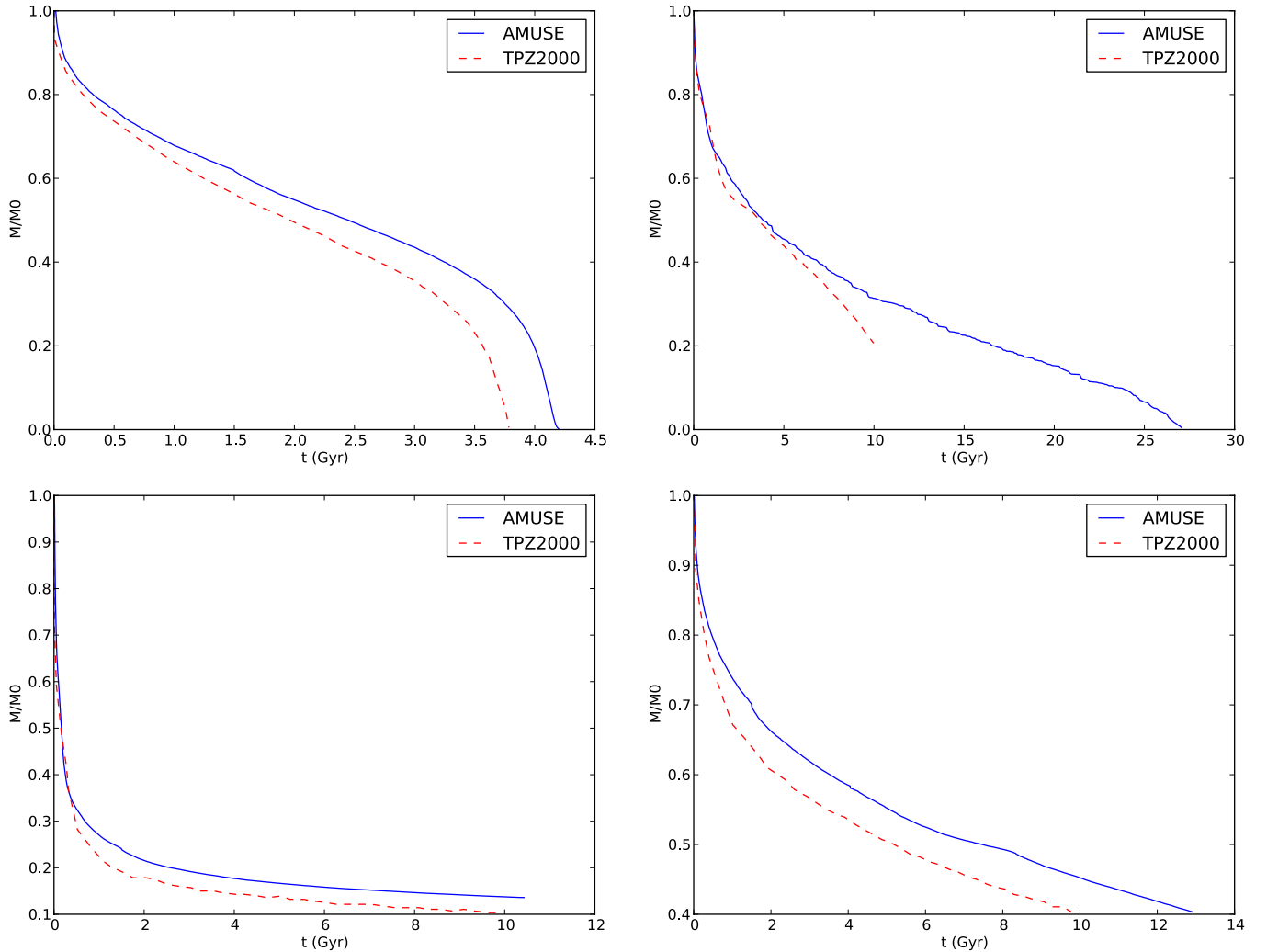


Figure 2. Evolution of the mass of a cluster of $N = 32,000$ stars, with initial cluster parameters $(W_0, \alpha, \text{family})$, from left to right: $(3, 2.5, 1)$, $(3, 2.5, 4)$, $(7, 1.5, 4)$, and $(7, 2.5, 1)$. The track produced by AMUSE is shown as the solid line (blue in the electronic version). The corresponding run result from Takahashi & Portegies Zwart (2000) (N -body model) is shown as the dashed line (red in the electronic version).

(A color version of this figure is available in the online journal.)

the former against the latter. Our initial goal was to see whether, and how well, AMUSE could replicate the TPZ N -body results.

In TPZ, curves for four choices of parameters are published. Using the notation $(W_0, \alpha, \text{family})$, they are: $(3, 2.5, 1)$, $(3, 2.5, 4)$, $(7, 1.5, 4)$, and $(7, 2.5, 1)$. We therefore generated initial conditions for each of these cases and ran them using AMUSE.

Figure 2 shows the result of our attempt to reproduce selected TPZ runs. The qualitative shapes of the curves are similar, but it is difficult to make any quantitative statement about the agreement or disagreement between the two sets of simulations.

An obvious suspect for these variances is the difference in stellar evolution recipes, which are known to be similar but not identical. TPZ used SeBa (Portegies Zwart & Verbunt 1996) to model stellar evolution, while we have used SSE. It is likely that small differences in the handling of main-sequence lifetime and remnant masses for short-lived O and B stars could alter the later evolution of the cluster due to early mass loss. This is not the only possibility. Section 4.5 reviews the variance introduced by the choice of stellar evolution model.

One known difference is in the handling of kicks in supernova formation. Our top-level AMUSE script ignores any kick prescriptions built into the stellar evolution codes. However, some of the models (for example, CW and SSE) leave a

zero-mass “remnant” for some initial stellar masses. This represents the detonation of the remnant during the supernova and the ejection of any remaining material. Our AMUSE script treats this as the star disappearing. Slight differences in these prescriptions are likely to contribute to variations seen between stellar evolution codes. It is unlikely, however, that this was the dominant source of variation.

This led us to consider the inherent spread in our results due to random variations in the initial conditions (individual star masses, positions, and velocities), as opposed to systematic differences between the integrators. We conducted multiple random realizations of each of our chosen initial parameter sets. Each parameter set (for example, $W_0 = 3$, $\alpha = 2.5$, family = 1) describes a continuous model, and we transform this model into a discrete mass spectrum and set of spatial and velocity data using a variable random number seed. We refer to a set of random realizations of a given parameter set as “formally equivalent.”

Figure 3 shows the results of 21 different realizations of three selected models. These models are the (W_0, α) combinations $(3, 1.5)$, $(7, 2.5)$, and $(3, 2.5)$ with all four families simulated for each case. This subset is 12 of the 16 possible combinations of W_0 , α , and family for the TPZ choices of these parameters.

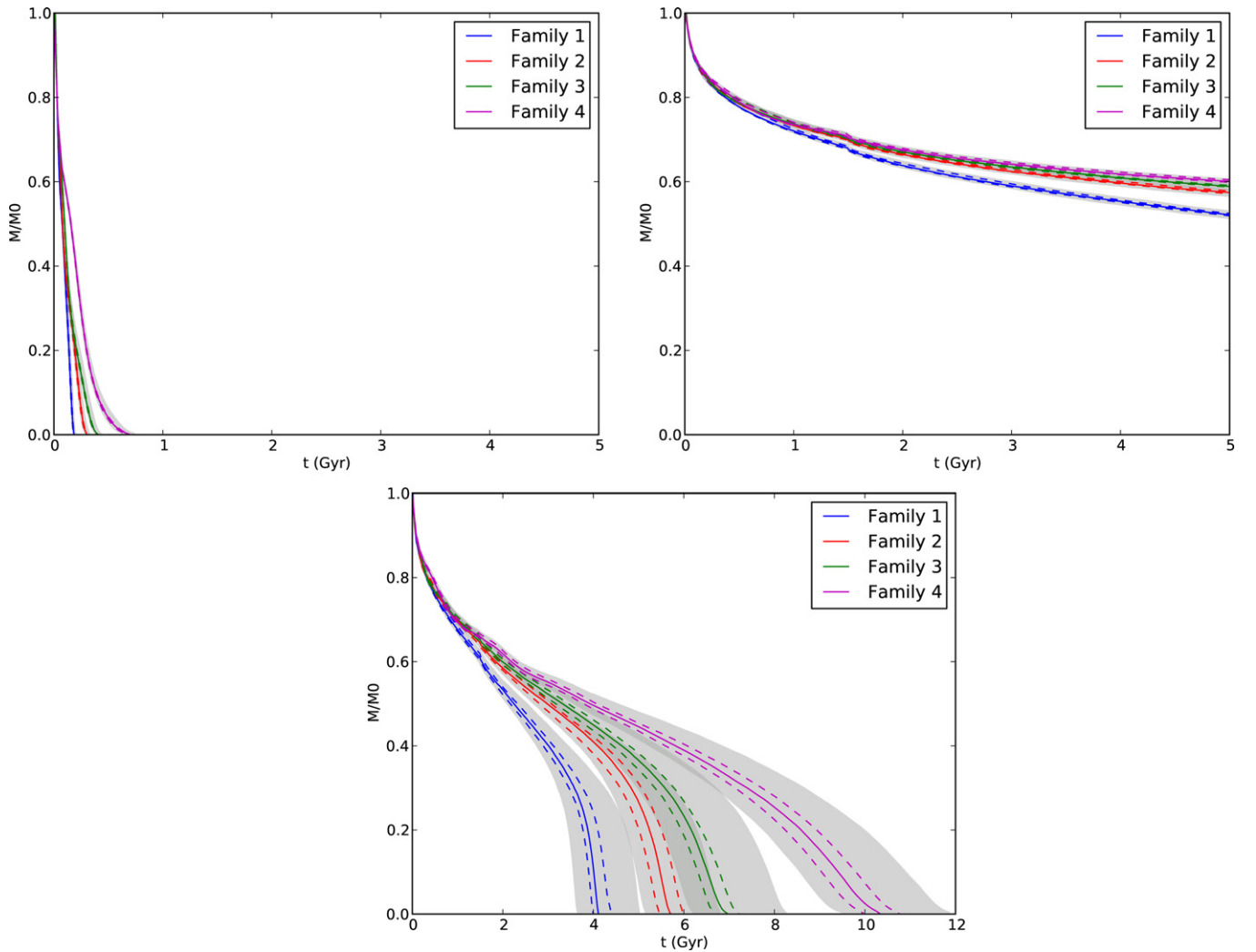


Figure 3. Evolution of the mass of a cluster of $N = 32,000$ stars, with initial cluster parameters (W_0, α) , from top-left clockwise: $(3, 1.5)$, $(7, 2.5)$, and $(3, 2.5)$. For each family, the solid line indicates the median value, the dashed lines indicated the 25th and 75th percentiles, and the shaded region indicates the total parameter space explored. This visualization is related to that in Ernst et al. (2011).

(A color version of this figure is available in the online journal.)

We do not show $(7, 1.5)$ as it is largely uninteresting: the cluster dissolves quickly, much like the $(3, 1.5)$ case. Furthermore, this is not a useful parameter set for validation because TPZ do not publish their cluster mass evolution curves for these combinations of parameters.

The variation due to randomness in the initial conditions is small, except for the case $W_0 = 3, \alpha = 2.5$, where the cluster dissolves but “slowly” relative to stellar evolution. “Slowly” refers to the fact that the initial mass loss driven by supernovae of O and B stars is not sufficient to disrupt the cluster. We explore this in further detail in Section 4.4. For $W_0 = 3, \alpha = 1.5$, the early stellar evolution loss due to massive stars dominates the cluster’s evolution, and it dissolves before differences in initial conditions can have much effect. Conversely, for $W_0 = 7, \alpha = 2.5$, the cluster is quite long-lived and the effects of initial condition variability are smoothed out over time.

4.2. Comparison with TPZ

Figure 4 also shows curves derived from TPZ’s STARLAB runs overplotted on an AMUSE simulation made using 21 runs done with PH4 and the SEBA module. Despite using the “same” stellar

evolution recipe, the results clearly do not agree. This is due to “drift” in the SEBA code since the TPZ paper was published in 2000. “Drift” refers to changes in the code underlying SEBA over time as the stellar evolution model is improved. Figure 5 shows one such change: the remnant mass kept by the code has changed in the 12 yr between the publication of TPZ and the current SEBA.

The lesson to simulators is clear: codes can change with time, and these changes can produce significant differences in the results of simulations.

4.3. Sources of Variance

We would like to know why some of the curves in Figure 3 have such large spreads in their tracks while others are constrained to a narrow area. In particular, the $W_0 = 3, \alpha = 2.5$ case behaves quite differently from the other cases. There are two obvious places in the discretization process where random variations might play a decisive role—the masses of the most massive stars, which explode early and can eject a significant amount of material from the cluster, and their locations, since mass ejected from the cluster center is much more destructive to

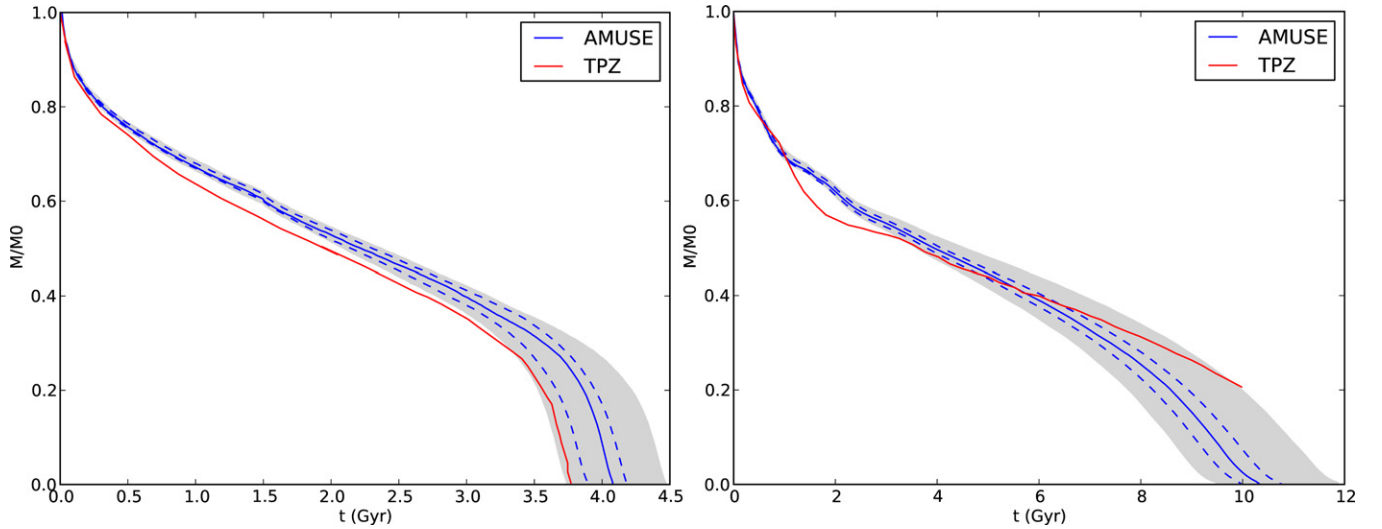


Figure 4. Evolution of the mass of a cluster of $N = 32,000$ stars, with initial cluster parameters $W_0 = 3$, $\alpha = 2.5$, family = 1 (left) or family = 4 (right). The range of variation in the AMUSE runs (made using PH4 and SEBA) is shown, with the solid lines indicating the median, the dashed lines the 25th and 75th percentiles, and the shaded area the entire envelope of the explored parameter space. A comparison is made to runs published by Takahashi & Portegies Zwart (2000) in red. (A color version of this figure is available in the online journal.)

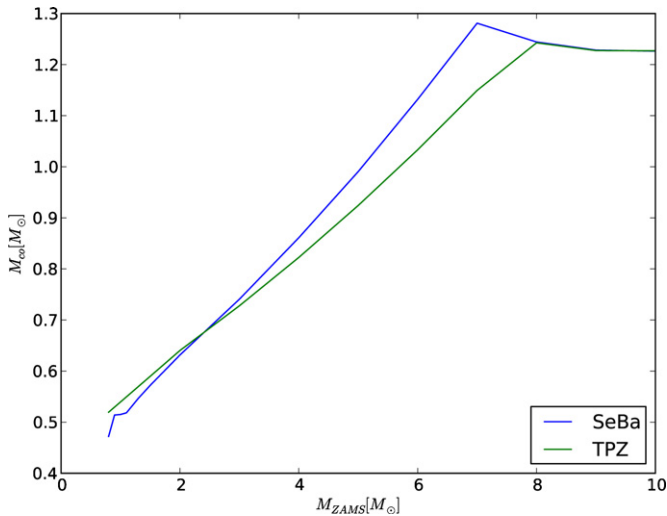


Figure 5. Change in the handling of remnants within SEBA between TPZ (published 2000) and the AMUSE SEBA module. Note that SEBA has been improved to better match the current understanding of stellar evolution.

(A color version of this figure is available in the online journal.)

the cluster than mass ejected from the outer regions (Vesperini et al. 2009; Pelupessy & Portegies Zwart 2012).

To separate these two effects, we explored the consequences of holding one of these “random” parameters constant and varying the other. For each of the two parameters, we ran 11 simulations with random choices of the other. Figure 6 illustrates the results. It is clear from the plot that varying the masses while holding the positions fixed has a larger effect than holding the masses constant and varying the positions. The mass spectrum effect is larger than the spatial effect by a factor of about two. These two effects are of the same order, and neither is negligible when comparing simulation results.

4.4. Types of Dissolution

We now focus our attention on those systems whose dissolution timescale is significantly shorter than that due solely to relaxation-driven mass loss, but longer than that due to early

stellar-evolution mass loss. These clusters’ lifetimes may be as long as several Gyr, but their dissolution should still be considered dynamical. This behavior was also observed by Chernoff & Weinberg (1990), Fukushima & Heggie (1995), and Takahashi & Portegies Zwart (2000). Fukushima & Heggie (1995) explained the mechanism of final disruption as a result of the loss of dynamical equilibrium within the cluster. We have examined the boundary between the ratios of the various timescales involved in order to map the boundary separating the slow and rapidly dissolving regimes.

There are three competing timescales in this simulation: the stellar mass loss timescale t_{SE} (which is of the order of a few tens of Myr at the beginning of the simulation and scales proportionally to t as the evolution of the cluster progresses), the dynamical timescale t_{dyn} (defined in Equation (6)), and the relaxation timescale t_{rh} (defined in Equation (4)). For a given choice of mass function slope α , the stellar mass loss timescale is fixed; it is an intrinsic property of the stars themselves and not of their dynamical phase space configuration.

The runs shown in Figure 3 show two modes of cluster dissolution: the flat slope of the long-lived ($W_0 = 7$, $\alpha = 2.5$) runs or the short-lived “ski jump” curve of the short-lived ($W_0 = 3$, $\alpha = 2.5$) runs. The ($W_0 = 3$, $\alpha = 1.5$) runs live extremely short lives. Nevertheless, it is clear from examining the family 4 curve that the “ski jump” is present in them too, and that they are more similar to ($W_0 = 3$, $\alpha = 2.5$) than to ($W_0 = 7$, $\alpha = 2.5$). It is clear that the lifecycle of massive stars plays a role in the disruption of the cluster in the short-lived “ski jump” cases. In the long-lived case of ($W_0 = 7$, $\alpha = 2.5$), the cluster survives the early stellar mass loss and evolves according to relaxation processes.

This inspired additional runs, conducted for $W_0 = 4, 5$, and 6 with $\alpha = 2.5$. A randomly realized selection of 10 runs was conducted for each family, and the results are plotted in Figure 7. The case ($W_0 = 5$, $\alpha = 2.5$, family = 4) is particularly interesting since it dissolves via dynamical processes while the remainder of the ($W_0 = 5$, $\alpha = 2.5$) models dissolve via relaxation processes.

Figure 8 is a schematic of the relationship between the three timescales for $N = 32,000$. We have subdivided the permitted

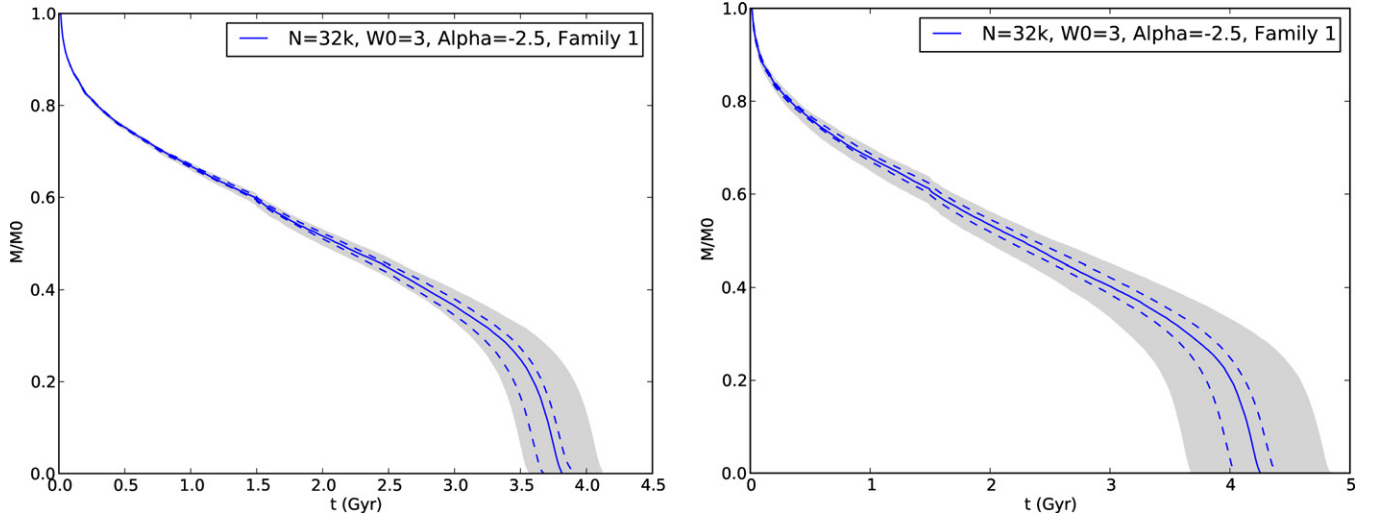


Figure 6. Evolution of the mass of a cluster of $N = 32,000$ stars, with initial cluster parameters $W_0 = 3$, $\alpha = -2.5$, family = 1. The range of variation seen in AMUSE runs is shown in blue, with the solid lines indicating the median, the dashed lines the 25th and 75th percentiles, and the shaded area the entire envelope of explored parameter space. The left-hand plot shows the result when the same mass list is applied to differing spatial positions, while the right-hand plot shows the result when the same spatial positions are matched to differing mass lists.

(A color version of this figure is available in the online journal.)

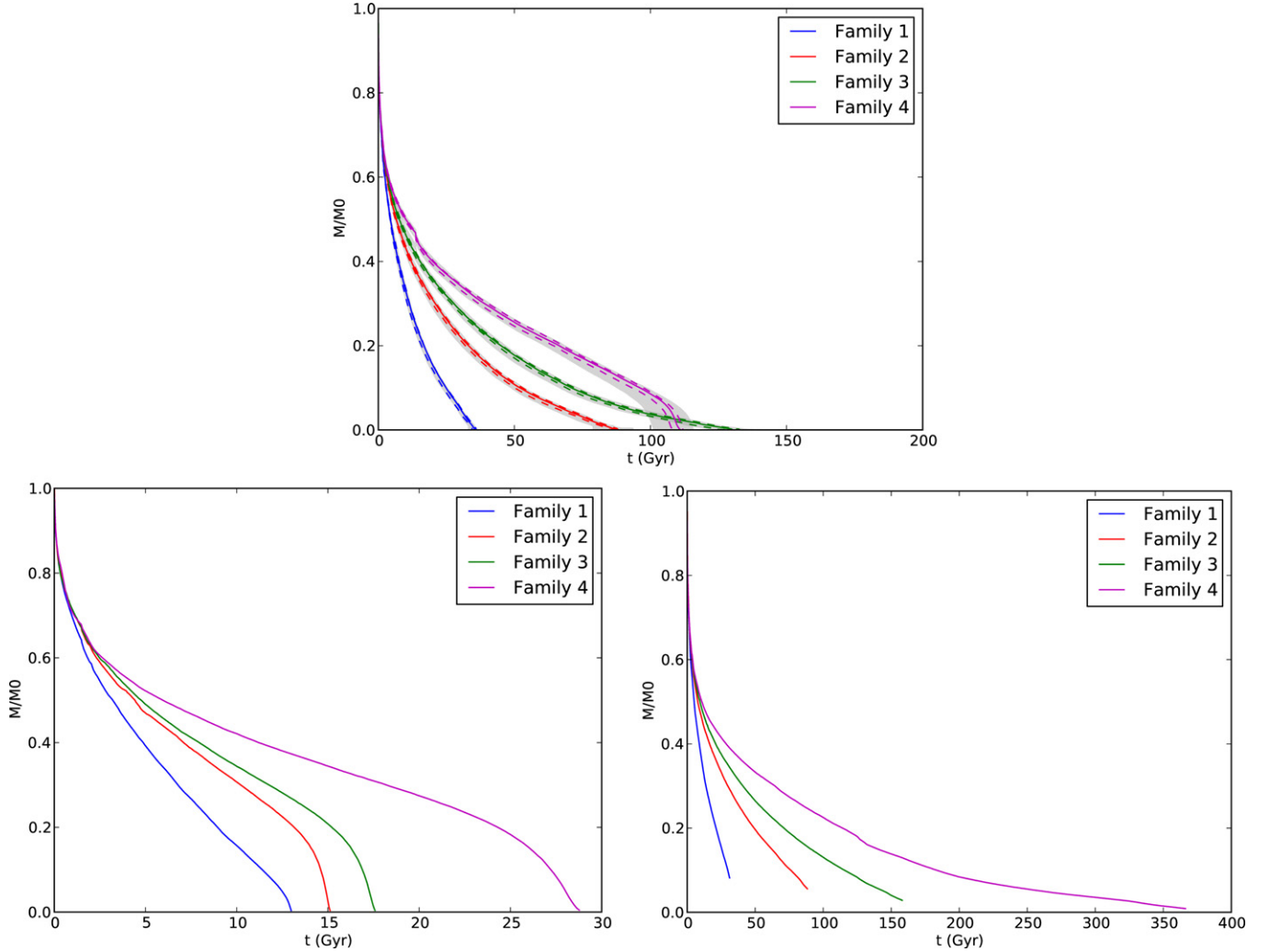


Figure 7. Evolution of the mass of a cluster of $N = 32,000$ stars, with initial cluster parameters (W_0, α) , from top-left clockwise: $(4, 2.5)$, $(6, 2.5)$, and $(5, 2.5)$. For each family, the solid line indicates the median value, the dashed lines indicates the 25th and 75th percentiles, and the shaded region indicates the total parameter space explored. The families are 1, 2, 3, and 4 in order from left to right. Note that family 4 in the $W_0 = 5$ case shows dynamical dissolution behavior while the other three families show relaxation dissolution behavior.

(A color version of this figure is available in the online journal.)

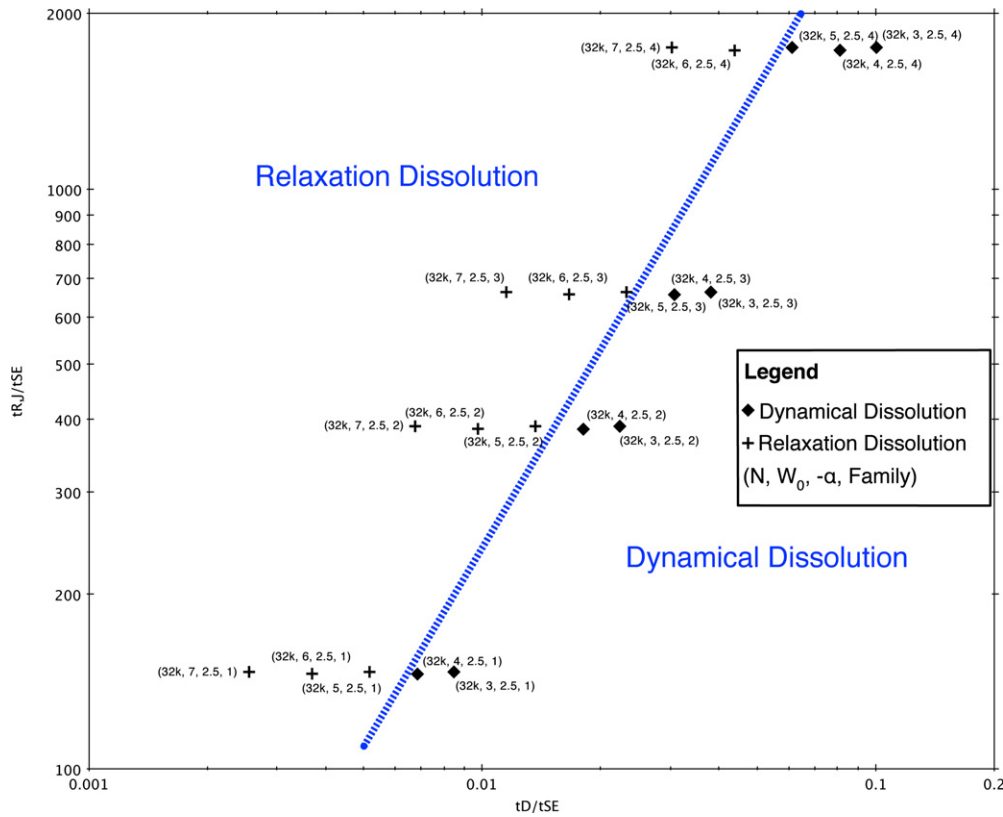


Figure 8. Space of combinations of the three fundamental star cluster timescales for $N = 32,000$. We identify two permitted regions where the death of the star cluster can be characterized as due to “disruption” (cluster dissolution triggered by the death of massive stars) or due to “relaxation” effects (due to the slow evolution of the cluster’s dynamical parameters). The label for each point is $(N, W_0, \alpha, \text{family})$. Dissolution results are denoted with the diamonds. Relaxation results are shown as the crosses.

(A color version of this figure is available in the online journal.)

region into two areas. In the rightmost region, the stellar mass loss timescale is much smaller than the dynamical timescale. This corresponds to the behavior seen in the $(W_0 = 3, \alpha = 1.5)$ runs: the cluster is rapidly disrupted by the death of massive stars early in its life. In this region, clusters dissolve due to disruption.

In the leftmost region, the opposite is true. In the leftmost region, as the ratio of the dynamical to stellar mass loss timescales decreases, so do the effects of stellar mass loss on the cluster structural evolution. For systems in this region (e.g., the family of models with $(W_0 = 7, \alpha = 2.5)$), the cluster dissolution is driven mainly by two-body relaxation mass loss. In our terminology, the cluster dissolves due to relaxation.

There is a small forbidden region near the horizontal axis that is not shown due to its small size. This region is forbidden because $t_{\text{rlx,CW}}$ cannot be smaller than t_{dyn} . Similar diagrams exist for other values of N and α .

The intermediate region (shown as a line in Figure 8) is an interesting case. We have seen that for $(W_0 = 3, \alpha = 2.5)$, the cluster dissolves due to disruption. However, in published plots such as Fukushige & Heggie (1995), there are cases within this region of cluster dissolution due to relaxation. We suspected that this was a change in behavior caused by a differing ratio of the relaxation time to the dynamical time. In order to control this ratio, we adjusted the number of stars N for the case $(W_0 = 3, \alpha = 2.5, \text{family} = 1)$.

The result is plotted in Figure 9. In this plot, it appears that $N = 2000$ (or $N = 1000$) could be either a relaxation or

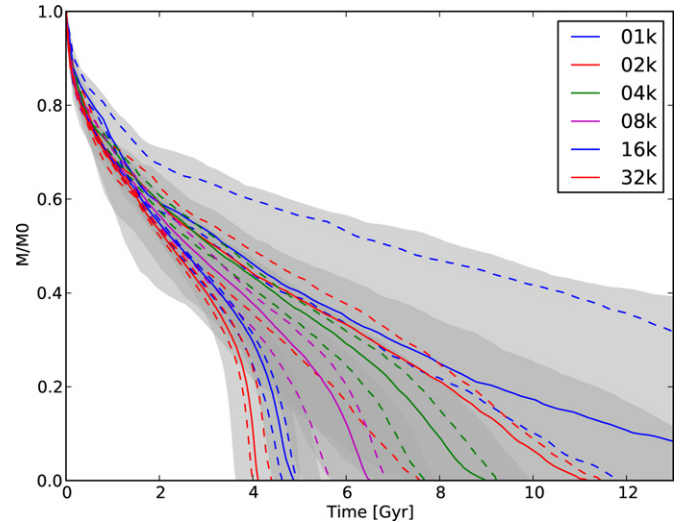


Figure 9. Star cluster mass loss tracks for clusters with variable number of stars N . Each N has 11 random realizations. The solid lines show the median of the runs, while the dashed lines show the 25th and 75th percentiles. All initial conditions are King models with $W_0 = 3, \alpha = 2.5, \text{family} = 1$.

(A color version of this figure is available in the online journal.)

disruption curve. However, if it were to be a relaxation curve, then we would expect to see a longer lifetime for the cluster than the $N = 1000$ case (compared, for example, to Figure 3).

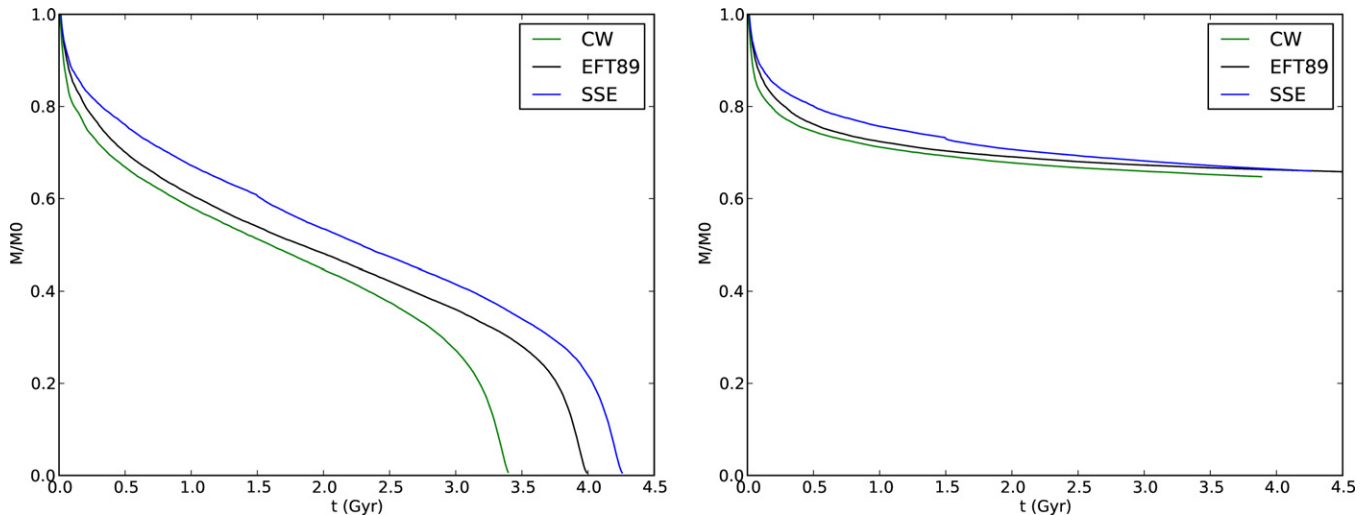


Figure 10. Evolution of the mass of a cluster of $N = 32,000$ stars, with initial cluster parameters $W_0 = 3$, $\alpha = -2.5$, family = 1 using different stellar evolution models. The left-hand frame shows the results if gravitational dynamics, stellar evolution, and a tidal cutoff are used. The right-hand plot shows the result of stellar evolution acting in isolation. The blue, green, and black lines correspond, respectively, to the stellar evolution prescriptions of Hurley et al. (2000; SSE), Chernoff & Weinberg (1990; VSSE), and Eggleton et al. (1989, 1990).

(A color version of this figure is available in the online journal.)

The $N = 2000$ lifetime is shorter than the $N = 1000$ lifetime, though, which places the $N = 2000$ case in the disruption region. A similar argument holds for the $N = 1000$ case. These small N runs are, in fact, more analogous to the ($W_0 = 3$, $\alpha = 1.5$) runs from Figure 3. In these cases, the cluster is dissolving in a small number of dynamical times. This set of runs therefore shows that all of the dissolutions for this set of parameters have a dynamical disruption character.

One should note that dynamical disruption does not have to be a rapid process relative to a Hubble time. While there are certainly cases of clusters being disrupted within a few tens of Myr, there are also dynamical disruption cases where the cluster survives for several Gyr.

4.5. Stellar Evolution Comparison

Figure 10 shows a single $N = 32,000$ model ($W_0 = 3$, $\alpha = 2.5$, family = 1) evolved using different stellar evolution models. AMUSE easily allows switching stellar evolution models in the same code. We used the SSE module, as used in the previous sections, as well as the idealized “VSSE” module described previously, and an implementation of the recipes given by Eggleton et al. (1989). All curves used were computed by AMUSE using PH4 for gravitational dynamics. Also included is a plot of the population synthesis (stellar evolution only) results for these initial conditions using the different stellar evolution models.

The small differences in the mass loss rates of the stars in the various models are amplified by the effects of gravitational dynamics and the removal of escapers to produce differences in the overall cluster lifetime of approximately 25%. The early supernovae experienced by O and B stars drive the cluster toward a faster dissolution because the relationship between the mass loss and shrinking tidal radius is self-reinforcing. When mass is lost, the tidal radius shrinks, which may leave some stars beyond the “edge” of the cluster, thus leading to additional mass loss and starting the cycle over again. These are systematic differences comparable in magnitude to the random run-to-run variations discussed in the previous section.

5. CONCLUSIONS AND FUTURE WORK

5.1. Scientific Results

In this paper, we examined whether or not AMUSE could be compared to published runs, in particular those of TPZ that reproduce prior work in CW. We also studied the source of variance between formally equivalent simulation runs and compared the effect of varying the stellar evolution model on the lifetime of a star cluster. The dissolution mechanism (disruption versus relaxation) for star clusters was explored.

Our AMUSE runs agree with Takahashi & Portegies Zwart (2000) (which agrees with Chernoff & Weinberg 1990), when all variances are taken into account. The sources of understood variances are the specific differences in the random realization of the mass spectrum, the random realization of the initial spatial distribution of stars, and the difference in remnant masses produced by the 2000 and 2012 versions of SeBa. These small differences are amplified by the dynamical interactions of stars over the cluster’s lifetime. It is not surprising that the remnant mass prescriptions have changed over the past 12 yr, as this has been a topic of active research in the community.

The direct comparison of stellar evolution codes has yielded an interesting result: prescriptions that differ by no more than a few percent in population synthesis studies can drive otherwise identical star cluster simulations to evolve at paces that differ by up to 25%. The cause is an amplification effect between mass loss due to stellar evolution and mass loss due to stars escaping the cluster. As the cluster loses mass, particularly due to early supernovae of O and B stars, the tidal radius of the cluster shrinks and more outlying stars are stripped from the cluster by the galaxy.

The use of random realization to generate initial conditions was also examined, and we found that formally equivalent (i.e., same W_0 , α , and CW family) initial conditions do not always follow the same evolutionary tracks. This effect is important in those cases where the star cluster does not dissolve rapidly (such as the loosely bound $W_0 = 3$ case with a top-heavy mass function of slope $\alpha = 1.5$) or remains tightly bound

over a Hubble time (such as $W_0 = 7, \alpha = 2.5$). Only in the intermediate regime (i.e., $W_0 = 3, \alpha = 2.5$) does this effect matter.

Variation in the evolution and dissolution time of a star cluster in the intermediate range is due primarily to the variation in masses of the most massive cluster members, but the effect of the differences in stellar spatial distribution due to random realization is also non-negligible. As had been expected, the population of O and B stars dominates the early cluster evolution because the effects of mass loss from supernovae are amplified as the cluster ages. The specific random realization of the spatial distribution also affects the overall cluster evolutionary track, but only at about half of the significance of the top end of the mass spectrum. This leads us to believe that the fraction of O and B stars alone is not sufficient to describe the variance they introduce; we must also consider the position within the cluster of those stars.

We have shown that there is a parameter-space boundary between King models that dissolve via dynamical processes and those that dissolve via relaxation processes. Figure 8 shows the location of this boundary in the space of the three relevant timescales: the stellar evolution mass loss timescale, the cluster's relaxation timescale, and the cluster's dynamical timescale.

5.2. Computational Method Results

The AMUSE framework shows promise as a new method of binding together domain-specific physics codes to form a larger simulation. Future work will be directed toward improving the abilities of this young framework relative to existing monolithic codes. Specifically, new dynamical modules have recently been added to handle close encounters between stars, as well as the formation, evolution, and destruction of binary and multiple stars. The addition of these modules will allow AMUSE to follow the evolution of a star cluster into core collapse and beyond.

In parallel to this development, the AMUSE team has also implemented modules to provide gas dynamics (generalized hydrodynamics, in fact) and radiative transfer processes. Using these modules, AMUSE should be able to perform production quality simulations including all of these components.

The modular structure of AMUSE facilitates comparison of physics modules and enables exploration of assumptions and approximations that is difficult or impossible with other simulation codes. We have used AMUSE to compare the effect of changing the stellar evolution prescription on an otherwise identical simulation, and used that to demonstrate that “small” differences between prescriptions are, in some cases, significant in the cluster's evolution. The ability to directly compare individual scientific codes within a multi-physics simulation is a novel property of a modular framework.

The interchangeability of modules benefits

1. the users of simulation codes, who may now select codes from a “menu” of choices;
2. those who interpret simulation results, who can now easily compare different implementations of the same underlying physics head-to-head; and

3. authors of new codes, who may focus their code on a single area of physics, knowing that it may be linked to a multi-physics environment easily, and that cross-validation of their work with existing codes can be accomplished by performing controlled experiments against established modules.

We thank Arjen van Elteren for his help with the new AMUSE interfaces. This work has been supported by NASA grants NNX07AG95G and NNX08AH15G, and by NSF grant AST0708299. AMUSE is developed at the Leiden Observatory, a faculty of Leiden University, and is funded by NOVA and NWO (grants VICI (No. 639.073.803), AMUSE (No. 614.061.608), and LGM (No. 643.200.503)). Part of the work was done while the authors visited the Center for Planetary Science (CPS) in Kobe, Japan, during a visit that was funded by the HPCI Strategic Program of MEXT. Another part was done while visiting the Lorentz Center. We are grateful to these centers for their hospitality.

REFERENCES

- Aarseth, S. J. 1999, *PASP*, **111**, 1333
Aarseth, S. J., & Heggie, D. C. 1998, *MNRAS*, **297**, 794
Casertano, S., & Hut, P. 1985, *ApJ*, **298**, 80
Chernoff, D. F., & Weinberg, M. D. 1990, *ApJ*, **351**, 121
Drost, N., Maassen, J., van Meersbergen, M. A. J., et al. 2012, arXiv:1203.0321
Eggleton, P. P., Fitchett, M. J., & Tout, C. A. 1989, *ApJ*, **347**, 998
Eggleton, P. P., Fitchett, M. J., & Tout, C. A. 1990, *ApJ*, **354**, 387
Ernst, A., Just, A., Berczik, P., & Olczak, C. 2011, *A&A*, **536**, 64
Fryxell, B., Olson, K., Ricker, P., et al. 2000, *ApJS*, **131**, 273
Fukushige, T., & Heggie, D. C. 1995, *MNRAS*, **276**, 206
Gaburov, E., Harfst, S.M., & Portegies Zwart, S. F. 2009, *NewA*, **14**, 630
Garcia, V. 2008, PhD thesis, Univ. Nice-Sophia Antipolis, Sophia Antipolis, France
Garcia, V., Debreuve, E., & Barlaud, M. 2008, in Proc. CVPR Workshop on Computer Vision on GPU, Anchorage, AK, USA (Red Hook, NY: IEEE via Curran Associates), 1
Harfst, S., Gualandris, A., Merritt, D., et al. 2007, *NewA*, **12**, 357
Harfst, S., Portegies Zwart, S. F., & McMillan, S. L. W. 2008, *AN*, **329**, 885
Hurley, J., Pols, O. R., & Tout, C. A. 2000, *MNRAS*, **315**, 543
Iben, I., & Renzini, A. 1983, *ARA&A*, **21**, 271
King, I. R. 1966, *AJ*, **71**, 64
McMillan, S. L. W., Portegies Zwart, S., van Elteren, A., & Whitehead, A. J. 2012, in ASP Conf. Proc. 453, *Advances in Computational Astrophysics: Methods, Tools, and Outcome*, ed. R. Capuzzo-Dolcetta, M. Limongi, & A. Tornambè (San Francisco, CA: ASP), 129
Miller, G. E., & Scalo, J. M. 1979, *ApJS*, **41**, 513
Pelupessy, F. I., & Portegies Zwart, S. 2012, *MNRAS*, **420**, 1503
Pelupessy, F. I., van Elteren, A., de Vries, N., et al. 2013, *A&A*, **557**, 84
Portegies Zwart, S. F., Makino, J., McMillan, S. L. W., & Hut, P. 1999, *A&A*, **348**, 117
Portegies Zwart, S. F., McMillan, S. L. W., & Gieles, M. 2010, *ARA&A*, **48**, 431
Portegies Zwart, S. F., McMillan, S. L. W., Harfst, S., et al. 2009, *NewA*, **14**, 369
Portegies Zwart, S. F., McMillan, S. L. W., Hut, P., & Makino, J. 2001, *MNRAS*, **321**, 199
Portegies Zwart, S. F., McMillan, S. L. W., van Elteren, A., Pelupessy, I., & de Vries, N. 2013, *CoPhC*, **184**, 456
Portegies Zwart, S. F., & Verbunt, F. 1996, *A&A*, **309**, 179
Salpeter, E. E. 1955, *ApJ*, **121**, 161
Springel, V., Yoshida, N., & White, S. D. M. 2001, *NewA*, **6**, 79
Takahashi, K., & Portegies Zwart, S. F. 2000, *ApJ*, **535**, 759
Vesperini, E., McMillan, S. L. W., & Portegies Zwart, S. F. 2009, *ApJ*, **698**, 615
Walker, D. W. 1994, *ParC*, **20**, 657